

» Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

» Katalog książek

- Katalog online
- Zamów drukowany katalog

» Twój koszyk

- Dodaj do koszyka

» Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

» Czytelnia

- Fragmenty książek online

» Kontakt

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl
© Helion 1991–2011

Tworzenie aplikacji iOS na urządzenia iPhone, iPod touch oraz iPad. Przewodnik dla projektantów serwisów WWW

Autor: [Kristofer Layon](#)

Tłumaczenie: Łukasz Suma

ISBN: 978-83-246-3331-9

Tytuł oryginału: [The Web Designer's Guide to iOS Apps:](#)

[Create iPhone, iPod touch, and iPad apps](#)

[with Web Standards \(HTML5, CSS3, and JavaScript\)](#)

Format: 170 × 230, stron: 264



Twórz atrakcyjne, zgodne ze standardami WWW aplikacje na urządzenia iPhone, iPod touch oraz iPad!

- Jak używać kodu HTML, stylów CSS i języka JavaScript w projektowaniu natywnych aplikacji?
- Jak korzystać z możliwości obiektowej platformy NimbleKit?
- Jak wizualizować, planować, projektować i dystrybuować aplikacje iOS?

Są z nami praktycznie wszędzie. Podczas zakupów, jazdy samochodem, w trakcie joggingu...

a nawet na basenie. Zmieniły nasz sposób pozyskiwania informacji, poszukiwania rozrywki i komunikacji ze światem. Dzięki urządzeniom takim jak iPhone czy iPad możesz w jednej chwili sfotografować niezwykłą scenę i rozesłać zdjęcie wszystkim znajomym, by potem wspólnie wymieniać na jego temat opinie, jednocześnie sprawdzając prognozę pogody i to, jak daleko jesteś od celu, do którego zmierzasz... Możliwości mobilnych aplikacji pracujących pod kontrolą iOS ogranicza dziś jedynie ludzka wyobraźnia! Jednak nic nie przemawia do niej tak, jak twarde dane: tylko do połowy 2010 roku sprzedanych zostało ponad 85 milionów urządzeń pracujących w tym systemie, a w iTunes App Store dostępnych było ponad 250 tysięcy aplikacji... pobranych ponad 15 miliardów razy!

Oto książka poświęcona projektowaniu atrakcyjnych, zgodnych ze standardami sieciowymi aplikacji działających pod kontrolą systemu iOS — aplikacji, które można udostępniać lub sprzedawać za pośrednictwem iTunes App Store. Podręcznik stanowi doskonałe wprowadzenie w świat kodu HTML, stylów CSS i języka JavaScript, z którymi zaprojektujesz niezawodne natywne aplikacje, działające na iPhone'ach, iPodach touch oraz iPadach. Dzięki niemu nauczysz się korzystać z platformy obiektowego C (o nazwie NimbleKit). Platforma ta jest kolekcją bibliotek elementów umożliwiających tworzenie programów za pomocą obiektowego języka C, bez konieczności samodzielnego pisania jakiegokolwiek kodu. Książka ta jest ponadto wszechstronnym przewodnikiem po kwestiach wizualizowania, planowania, tworzenia i dystrybucji aplikacji iOS oraz szczegółową instrukcją projektowania rozmaitych aplikacji.

◀ **Kristofer Layon** ▶

PRZEWODNIK DLA PROJEKTANTÓW SERWISÓW WWW

Tworzenie aplikacji



na urządzenia iPhone,
iPod touch oraz iPad



Twórz atrakcyjne, zgodne ze standardami WWW aplikacje na urządzenia iPhone, iPod touch oraz iPad!

Jak używać kodu HTML, stylów CSS i języka Java Script w projektowaniu natywnych aplikacji?

Jak korzystać z możliwości obiektowej platformy NimbleKit?

Jak wizualizować, planować, projektować i dystrybuować aplikacje iOS?

Helion



Spis treści

Podziękowania	9
O autorze	11
Wstęp	13
Rozdział 1. Wielkie znaczenie niewielkich rozmiarów	17
Mobilna magia i komputery kieszonkowe	18
Treść — i kontekst — są wszystkim	19
Aplikacje mobilne ≠ aplikacje biurkowe	21
Magia transformacji	22
Projektowanie zaczyna się od ludzi, a kończy na kodzie	25
Podsumowanie	27
Rozdział 2. Zakładanie studia projektowania aplikacji	29
Uzyskiwanie Apple Developer ID	30
Pobieranie i instalowanie pakietu iOS SDK	35
Pobieranie i instalowanie pakietu NimbleKit	37
Podsumowanie	39
Rozdział 3. Podstawy iOS SDK	41
Rozpoczynanie nowego projektu Xcode	42
Testowanie i budowanie wykonywalnego pliku aplikacji	55
Podsumowanie	65
Rozdział 4. Interfejs użytkownika i funkcjonalność iOS	67
Czym jest pasek stanu?	68
Implementowanie paska tytułu	71
Projektowanie z wykorzystaniem pasków kart	72
Nawigacja z wykorzystaniem widoków tabeli	75
Podsumowanie	82

Rozdział 5. Koncentracja na treści — tekst i grafika	83
Strukturyzowanie tekstu	84
Dołączanie treści społecznościowych	90
Praca z obrazami	95
Podsumowanie	103
Rozdział 6. Koncentracja na treści — mapy	105
Metoda pierwsza: użycie przycisku UIButton	107
Metoda druga: formatowanie przycisku HTML	114
Przypadek iPada	119
Podsumowanie	123
Rozdział 7. Koncentracja na treści — dźwięk	125
Odtwarzanie dźwięku za pomocą kodu HTML5	126
Dołączanie dźwięku za pomocą elementu <code>HTML5AudioPlayer</code>	128
Podsumowanie	132
Rozdział 8. Koncentracja na treści — wideo	133
Dostarczanie treści wideo do urządzenia iPad za pomocą kodu HTML5	134
Dostarczanie treści wideo za pomocą elementu <code>HTML5VideoPlayer</code>	140
Podsumowanie	146
Rozdział 9. HTML5 i CSS3	149
Odkrywanie dodatkowych elementów języka HTML5	150
Więcej możliwości projektowych dzięki kodowi CSS3	155
Podsumowanie	178
Rozdział 10. Inne platformy do tworzenia aplikacji mobilnych	179
Emulowanie działania systemu iOS za pomocą rozwiązań PhoneGap i jQueryTouch	180
Tworzenie natywnych aplikacji za pomocą platformy Titanium Mobile	186
Tworzenie aplikacji WWW za pomocą platformy Sencha Touch	189
Podsumowanie	191

Rozdział 11. Promowanie aplikacji	193
Kim jesteś — wybór tożsamości w App Store	194
Korzystanie z możliwości marketingowych oferowanych przez firmę Apple	197
Tworzenie własnych kanałów marketingowych związanych z aplikacją	201
Podsumowanie	210
Rozdział 12. Próbną instalacja i dystrybucja aplikacji	211
Korzystanie z iOS Dev Center	212
Korzystanie z iTunes Connect	221
Dodawanie aplikacji i zarządzanie nimi	223
Podsumowanie	230
Dodatek A. Dodatkowe wytyczne dotyczące projektowania	231
Strategia tworzenia treści	232
Planowanie aplikacji	237
Funkcjonalność aplikacji	242
Przypisy końcowe	249
Skorowidz	251

Rozdział 1.

Wielkie znaczenie niewielkich rozmiarów

*Zatem... dlaczego napisałem tę książkę?
Czyż nie ma książek o tworzeniu aplikacji
dla iPhone'ów, iPodów touch i iPadów?*

Rzeczywiście, jest już kilka książek na ten temat i wszystkie one są bardzo pouczające. Tę napisałem jednak dla bardzo szczególnej grupy czytelników.

W skrócie: dla ludzi podobnych do mnie.

Napisałem więc książkę, którą bardzo chciałem mieć na półce jakieś dwa lata temu, w czasie gdy zaczynałem poznawać sposób projektowania aplikacji dla iPhone'a. Wtedy udało mi się znaleźć jedynie publikacje traktujące o programowaniu w obiektowym C lub poświęcone temu, jak wykorzystywać bardzo specyficzne funkcje i możliwości oferowane przez urządzenia iPhone i iPod touch.

Nie mam nic przeciwko programowaniu w obiektowym C. Po prostu sam raczej wołałbym tego nie robić. I choć pozostaję pod wielkim wrażeniem możliwości mobilnych urządzeń firmy Apple, nadal jestem projektantem: tym, co motywuje mnie do pracy, nie jest sama technologia, lecz pragnienie, aby ułatwić komunikację ludziom i organizacjom.

Jeśli zatem jesteś projektantem, którego bardziej bawi praca z ludźmi niż zmaganie się z technologią i rozwiązywanie problemów niż eksperymentowanie z funkcjami, to właśnie znalazłeś odpowiednią książkę, ponieważ zamierzam zająć się w niej projektowaniem aplikacji iOS, skoncentrowanym na potrzebach użytkownika.

Mobilna magia i komputery kieszonkowe

Wraz z upowszechnianiem się iPhone'ów i innych smartfonów znacznie wzrosło zapotrzebowanie na dobrze zaprojektowane treści mobilne. Wszyscy mogliśmy być świadkami publikacji kolejnych zdumiewających danych statystycznych:

- ▶ do połowy 2010 roku zostało sprzedanych ponad 85 milionów urządzeń pracujących pod kontrolą iOS,
- ▶ w iTunes App Store dostępnych było ponad 250 tysięcy aplikacji,
- ▶ iTunes odnotowało 15 miliardów pobrań aplikacji.

Osobiście doświadczyłem zmieniających pogląd na świat skutków wynikających z możliwości uzyskania dostępu do treści w niemal każdym miejscu i w czasie, gdy tylko są one potrzebne: podczas zakupów, pracy, treningu, a nawet w czasie jazdy wyciągiem krzesełkowym na stoku narciarskim. Jeśli tylko nie jestem na plaży lub w wodzie, zwykle mam przy sobie swojego iPhone'a. Mogę odpowiadać na pytania. Mogę przyjrzeć się czemuś, co sfotografowałem wcześniej jako punkt odniesienia lub obiekt do porównania [często używam też aplikacji Aparat (ang. *Camera* do „robienia notatek”), sprawdzić, jak daleko jestem od celu, zapoznać się z aktualnym stanem pogody. Lista możliwych działań praktycznie nie ma końca.

W gruncie rzeczy myślę wręcz, że nazwa iPhone'a jest nieco myląca, sugeruje bowiem, że chodzi tu o urządzenie będące telefonem wyposażonym w dodatkowe funkcje. A tak naprawdę jest to ściśle zintegrowany z siecią komputer kieszonkowy, z którym możesz robić rozmaite rzeczy:

- ▶ zabrać ze sobą, gdzie tylko zechcesz,
- ▶ używać, gdy tylko zajdzie taka potrzeba,
- ▶ dostosowywać do swoich wymagań poprzez kupowanie i instalowanie samodzielnie wybranych aplikacji.

Możemy zatem myśleć o iPhone'ie jako o telefonie (rysunek 1.1), ponieważ funkcjonuje na nim aplikacja Telefon (ang. *Phone*). Jest to jednak tylko jedna z wielu aplikacji zapewniających dostęp do treści i połączenia z siecią oraz oferujących możliwość korzystania z różnych sprzętowych i programowych funkcji urządzenia, które pozwalają rozwiązywać rozmaite problemy lub uzyskiwać dostęp do informacji akurat wtedy, gdy są Ci potrzebne, i tam, gdzie ich naprawdę potrzebujesz.



Rysunek 1.1. To jest prawdziwy telefon!
(Podczas gdy Telefon jest tylko jedną z wielu aplikacji działających na iPhone'ie)

Treść — i kontekst — są wszystkim

Dlaczego zatem koncentruję się tu na sprawach, które powinny już być dość oczywiste? Ponieważ staram się naprawdę wyraźnie podkreślić to, co ważne, i wskazać Ci sposób, w jaki powinieneś myśleć o projektowaniu aplikacji.

Aby kontynuować ten proces, przyjrzyj się tym dwóm datom:

9 stycznia 2007

oraz

25 maja 2010

Rozpoznajesz je? Obydwie mają ogromne znaczenie dla sposobu, w jaki powinniśmy myśleć o aplikacjach iOS.

9 stycznia 2007 to data przeistoczenia się firmy Apple Computer, Inc. w Apple, Inc. A 25 maja 2010 to dzień, w którym przedsiębiorstwo Apple, Inc. stało się najdroższą firmą technologiczną na świecie. Miało to miejsce trzy lata po odrzuceniu słowa „komputer” z nazwy korporacji.

Przemianę tę najlepiej podsumowała gazeta „New York Times”:

„Najważniejszy produkt technologiczny nie znajduje się już na twoim biurku, lecz raczej mieści się w twojej dłoni” (27 maja 2010).

Firma Apple dostrzegła tę tendencję już w początkach roku 2007, gdy postanowiła zmienić swoją nazwę. Być może miało to związek z faktem, że na horyzoncie pojawił się już wówczas iPhone, który na dobre miał zagościć na rynku w lecie tego samego roku.

Jednak to nie sam iPhone wiosną 2010 roku uczynił Apple największą firmą technologiczną na świecie. Proces ten rozpoczął się już w roku 2001, gdy na rynek wszedł iPod, i nabrał rozpędu w roku 2003, gdy udostępniono iTunes. Zwróć uwagę na fakt, że chociaż firma nie zaprzestała wtedy produkcji pełnowymiarowych komputerów, zaczęła oferować też o wiele mniejsze. To zaś pomogło wprowadzić technologię komputerową w nasze życie o wiele głębiej, niż kiedykolwiek miały szansę zrobić to komputery biurkowe czy laptopy.

Wprowadzając na rynek nową, niewielką platformę komputerową iPod, firma Apple zrobiła znacznie większy krok niż ten, który pozwolił jej pokonać dystans między komputerami Apple II i Mac. Na tym pierwszym etapie ewolucji przedsiębiorstwo spopularyzowało graficzny interfejs użytkownika (ang. *Graphic User Interface* — GUI), czyli wizualną reprezentację biurka, a także mysz w roli urządzenia wejściowego, dzięki czemu udało się osiągnąć zupełnie nową jakość, która obecnie już na stałe zagościła w świecie komputerów osobistych. W tej chwili większość użytkowników korzysta ze swoich komputerów i komunikuje się z nimi

bez konieczności posługiwania się ich wewnętrznymi językami (co oznacza, że nie muszą już oni programować komputerów, aby osiągnąć wymagane efekty).

Najbardziej niesamowite w iPodzie jest to, że jego możliwości w porównaniu do Maca są naprawdę mocno ograniczone. I na tym nie koniec. Apple nie tylko pozbawiła go myszy, ale także biurka. Firma wyposażyła iPoda w niewiarygodnie mały ekran i uniemożliwiła tworzenie jakichkolwiek treści bezpośrednio za pomocą samego urządzenia (rysunek 1.2). Wzięła więc na warsztat komputer, pozbawiła go większości możliwości, a następnie zmniejszyła i w maksymalnym stopniu uprościła. Na dodatek uczyniła go urządzeniem kompletnie bezproduktywnym, a co za tym idzie — zupełnie nieprofesjonalnym. Zastosowano tu zatem dokładnie odwrotną strategię, niż miało to miejsce w latach 90. minionego wieku w przypadku komputerów osobistych.



Rysunek 1.2. Oryginalny iPod — uproszczenie i zmniejszenie komputera w stopniu nieosiągalnym nawet dla Maców

Jaki był efekt tej drastycznej redukcji mocy obliczeniowej, możliwości i rozmiaru?

Wyniki sprzedaży poszybowały w kosmos i sprawiły, że firma odniosła niesamowity sukces rynkowy.

Przemiana ta poszła tak daleko, że nie myślimy już nawet o iPodzie jako o małym komputerze. Traktujemy go raczej jak przenośne, praktyczne i bardzo łatwe w użyciu urządzenie do odtwarzania muzyki, przeglądania nowinek i wiadomości, odsłuchiwania książek dźwiękowych oraz przeglądania zdjęć, a nawet oglądania filmów i programów telewizyjnych.

Treści te nie mają charakteru biznesowego. Te treści mają charakter codzienny i prywatny.

Co ciekawe, Apple nie wywołała takiego technologicznego trzęsienia Ziemi, wprowadzając na rynek swoje urządzenia działające pod kontrolą systemu iOS. Oferując iPhone'a w roku 2007, firma raczej tylko ponownie zapewniła dostęp do kilku najważniejszych funkcji, z których

zrezygnowała wcześniej, wykonując niewiarygodnie wielki krok od pełnowymiarowego komputera do małej platformy iPod. Przywrócono połączenie z internetem, do którego przyzwyczailiśmy się już podczas używania naszych komputerów osobistych i laptopów, ponownie zapewniono też możliwość wprowadzania danych za pomocą klawiatury (ekranowej) po wcześniejszym ograniczeniu tej możliwości do operacji wykonywanych za pomocą kółka sterującego iPodą.

A tak, nie należy zapominać jeszcze o czymś... iPhone został wyposażony w aplikację Telefon!

Co ważniejsze, urządzenia iOS znacznie rozszerzają koncepcję codziennych treści. Wiadomości ze świata mogą być teraz naprawdę aktualne, muzyka może być odtwarzana na żywo, a informacje mogą nawet obejmować plan rozgrywek małej ligi piłkarskiej. Albo adres restauracji, zaprezentowany na mapie wraz z wygodnymi wskazówkami na temat dojazdu z bieżącego miejsca.

Wszystko to oznacza, że projektowanie aplikacji dla tych urządzeń powinno się zaczynać od odpowiedniego skupienia się na codziennych treściach i życiowych potrzebach osób, które będą ich używać, a także na problemach, z którymi na co dzień się stykają.

Aby projektować aplikacje spełniające te wymagania, musimy stąpać twardo po ziemi.

Aplikacje mobilne ≠ aplikacje biurkowe

Musimy też jednak sprawić, aby były one proste.

Myślenie o mobilnych treściach z perspektywy codziennych treści umożliwia Ci skupienie się na właściwym kontekście Twojej pracy projektowej, który określa to, jak i kiedy ludzie poszukują informacji. Następnie powinniśmy się skoncentrować na tym, w jaki sposób ludzie korzystają z urządzeń mobilnych.

Wiele książek i prezentacji na temat projektowania aplikacji koncentruje się na wykorzystaniu specyficznych funkcji (uczą one na przykład, jak sprawić, aby telefon wibrował!). I z pewnością w przypadku dużego zespołu programistów pracujących nad skomplikowaną aplikacją zgłębianie konkretnych zagadnień technicznych i szczegółów związanych z działaniem sprzętu może się okazać sensowne. Książka ta została jednak napisana z perspektywy ludzkiej, w związku z czym ludzie i używane przez nich treści grają w niej pierwszoplanowe role.

Gdy ludzie korzystają z mobilnych treści, z pewnością nie koncentrują się na tym, czy ktoś, kto tworzył pośredniczące oprogramowanie, wykorzystał określoną funkcję urządzenia. Jeśli jednak projektanci aplikacji pominęli jakieś możliwości lub zaimplementowali je w niewłaściwy sposób, użytkownicy bez wątpienia natychmiast to zauważą. Skupienie się na treści i ludzkich potrzebach powinno pozwolić nam wykorzystać właściwe działania

i funkcje w odpowiedni sposób. Wydaje mi się to o wiele lepsze niż uczenie się najpierw, jak implementować określone funkcje, a następnie zastanawianie, jak wpleść je w tworzone aplikacje.

Projektowanie aplikacji przeznaczonej dla urządzenia mobilnego różni się znacznie od projektowania witryny, która będzie przeglądana za pomocą komputera wyposażonego w pełnowymiarowy monitor. Coraz częściej korzystamy z komputerów osobistych lub nawet laptopów — głównie w pracy lub w celach edukacyjnych. (Oczywiście sytuację tę dramatycznie zmienił internet, ale daj mi jeszcze chwilę). Oznacza to, że miejsca, w których zwykliśmy korzystać z komputerów, ograniczają się obecnie do biur, szkół i domów. Laptopy mogą być rzecz jasna zabierane w wiele różnych miejsc i niektórzy z nas noszą je ze sobą niemal wszędzie, większość osób używa ich jednak raczej w jednej lokalizacji.

Porównaj to ze sposobem korzystania z iPhone'a, iPoda touch lub iPada. W zależności od tego, jakim rodzajem i modelem urządzenia dysponujesz, możliwości związane z połączeniami mogą się różnić, mimo to jednak — mając na względzie współczesną wszechobecność sieci bezprzewodowych — posiadanie jednego z tych komputerów przenośnych w swojej kieszeni (choć to raczej przenośnia w przypadku posiadaczy iPadów — chyba że mają oni **naprawdę** wielkie kieszenie!) oznacza, że możesz korzystać z niego w drodze. To zaś z kolei znaczy, że kontekst może się często różnić, a powód używania urządzenia jest prawdopodobnie **zupełnie inny**, niż ma to miejsce w przypadku standardowego komputera. Przeglądanie przepisu kulinarnego w celu sprawdzenia listy składników raczej nie jest zadaniem typowo biznesowym, chyba że jesteś kucharzem lub zaopatrujesz restaurację. Bierzesz swojego iPhone'a lub iPoda touch ze sobą na zakupy spożywcze, ponieważ jesteś człowiekiem, jesteś głodny i musisz rozwiązać ten szczególny problem od ręki.

I nie jest to problem o charakterze technicznym.

Najważniejszą rzeczą, którą należy zapamiętać, jest to, że projektowanie treści dla urządzeń Apple działających pod kontrolą iOS różni się od projektowania treści wyświetlanych za pomocą przeglądarki internetowej na dużym ekranie. I chodzi tu o coś więcej niż sam kontekst. W przypadku pełnowymiarowego komputera przeglądarka dostosowuje się nieco do wyświetlanej treści, jednak w rzeczywistości nie myślimy o tym w taki sposób. Przeglądarka prezentuje nam stronę WWW i nadal znajduje się na naszym komputerze, a my w gruncie rzeczy nawet nie dostrzegamy zachodzących w niej zmian.

Magia transformacji

Oto kolejna kluczowa cecha wyróżniająca urządzenia działające pod kontrolą iOS — sposób dostarczania treści, zwłaszcza w natywnych aplikacjach zaprojektowanych w celu obsługi szczególnych potrzeb komunikacyjnych lub zastosowań, **staje się** rzeczą, którą projektujemy.

Jeśli Cię to nie przekonuje, porównaj powierzchnię zajmowaną przez ekran do obszaru przeznaczanego na pozostałe elementy i kontrolki w przypadku dwóch różnych rodzajów urządzeń elektronicznych.

iPhone, iPod touch oraz iPad są do siebie bardzo podobne w pewnej kwestii: ich powierzchnia to w głównej mierze ekran. Od przodu wszystkie trzy urządzenia w mniej więcej 95 procentach pokrywają wyświetlacze, podczas gdy w przypadku laptopa monitor stanowi mniej niż 50 procent powierzchni obudowy, ponieważ pozostałą jej część zajmuje klawiatura, inne elementy oraz puste płaszczyzny (rysunek 1.3). To zasadnicza różnica. Nawet najbardziej wciągająca strona internetowa wyświetlana na ekranie laptopa jest widoczna **na** laptopie, prawda? Klawiatura, gładzik, powierzchnia przeznaczona na podparcie nadgarstków oraz ramka otaczająca ekran nie znikają. W dalszym ciągu mają wpływ na nasz odbiór treści i utrzymują nas od niej z daleka.



Rysunek 1.3. Nawet w przypadku laptopa w oczu rzuca się przede wszystkim klawiatura, gładzik i ramka. Ekran to mniej niż 50 procent całkowitej powierzchni obudowy urządzenia

Jednak gdy tylko uruchomisz aplikację na urządzeniu iOS, całe ono wydaje się przekształcać, co zawdzięcza sposobowi, w jaki zostało zaprojektowane — główną rolę gra tu wyświetlacz oraz obsługa gestów i dotyku. W przypadku niektórych aplikacji wydaje się to bardziej oczywiste niż w innych, pomyśl jednak, jak mocno iPhone zaczyna przypominać telefon, gdy działa na nim aplikacja Telefon. Urządzenie niemal znika: nagle trzymamy w dłoni lśniąca klawiaturę telefonu z wbudowaną listą danych kontaktowych. Dokładnie tak samo jest w przypadku aplikacji Mapy (ang. *Maps*): w gruncie rzeczy nie chodzi tu o mapę znajdującą się w urządzeniu — to aplikacja pomaga mu **stać się mapą**.

Właściwe zrozumienie tego efektu transformacyjnego ma kluczowe znaczenie dla naszego sposobu projektowania aplikacji działających na tych urządzeniach. Musimy zachować szczególną ostrożność przy projektowaniu interfejsu użytkownika (ang. *User Interface* — UI).

Powinniśmy nauczyć się odpowiednio traktować natywne kontrolki iOS Apple i właściwie rozpoznawać sytuacje, gdy należy projektować własne elementy UI, które będą bezpośrednio obsługiwały potrzeby komunikacyjne tworzonej aplikacji. Zwłaszcza że — w niektórych przypadkach — rola elementów interfejsu bywa większa, niż wynika to tylko ze spełnianych przez nie funkcji, ponieważ znaczenie może mieć też kwestia marki.

Bardzo istotna jest też świadomość tego, **jak** ważne jest w tej sytuacji myślenie o projekcie. Ponieważ nie zgodziłbym się ze stwierdzeniem, że nieosiągnięcie pożądanego celu w przypadku natywnych kontrolki UI Apple bądź też kontrolki związanych z treścią lub marką odbiega znacząco od napisania klasówki i otrzymania zaledwie oceny dobrej zamiast celującej.

Niestety iluzja transformacji urządzenia w coś zupełnie innego jest o wiele bardziej ryzykowna niż to: minięcie celu nawet w niewielkim stopniu nie powoduje jedynie uzyskania dobrej aplikacji zamiast doskonałej. Zbyt łatwo może bowiem skutkować otrzymaniem aplikacji wybitnie nieprzekonującej lub nawet denerwującej.

Wszystko to może Ci się teraz zacząć wydawać dość skomplikowane. Dobrze zaprojektowana aplikacja powinna działać na wiele zmysłów, aby doskonale spełniać oczekiwania użytkownika, niezależnie od tego, gdzie się on znajduje, korzystając z natywnych elementów kontrolnych interfejsu użytkownika iOS Apple, gdy okazuje się to właściwe, **oraz** zawierać specyficzne dla treści lub marki szczegóły UI, gdy z kolei właśnie to jest odpowiednie. Na czym zatem polega to magiczne zaklęcie, dzięki któremu jest to możliwe? I jak możemy w wystarczającym stopniu skupić się zarówno na treści, jak i interfejsie użytkownika, aby udało się osiągnąć całościowy sukces?

Odpowiedź jest oczywiście taka, że nie istnieje jedno takie zaklęcie. Lecz na szczęście projektanci WWW mają już pewne doświadczenie w odpowiadaniu na potrzeby klientów i użytkowników, dobrze znają też metody projektowania zgodnego z wytycznymi korporacyjnymi dotyczącymi stylu. Aby wdrożyć to wartościowe doświadczenie uzyskane przy projektowaniu na potrzeby WWW w proces tworzenia aplikacji iOS, musisz zaznajomić się ze standardami, szczegółami i zaleceniami dotyczącymi natywnego interfejsu użytkownika, zdefiniować wymagania swojego projektu (niezależnie od tego, czy będzie to dzieło na własny użytek, czy też projekt klienta lub pracodawcy) oraz określić najlepszy sposób zaprojektowania jej pod kątem wymagań urzędzeń pracujących pod kontrolą systemu iOS.

W książce tej będę kierował się tym paradygmatem i pokażę Ci, jak wykorzystać Twoje doświadczenie projektowe oraz umiejętności związane z tworzeniem na potrzeby WWW w projektowaniu aplikacji iOS. Przedstawię również pewne techniki, które pozwolą Ci opracować te projekty bez konieczności pisania własnego kodu w obiektywnym C.

Projektowanie zaczyna się od ludzi, a kończy na kodzie

Z pewnością zastanawiasz się teraz, czy miałeś rację, myśląc, że wszystkie te aplikacje muszą być pisane w obiektowym C.

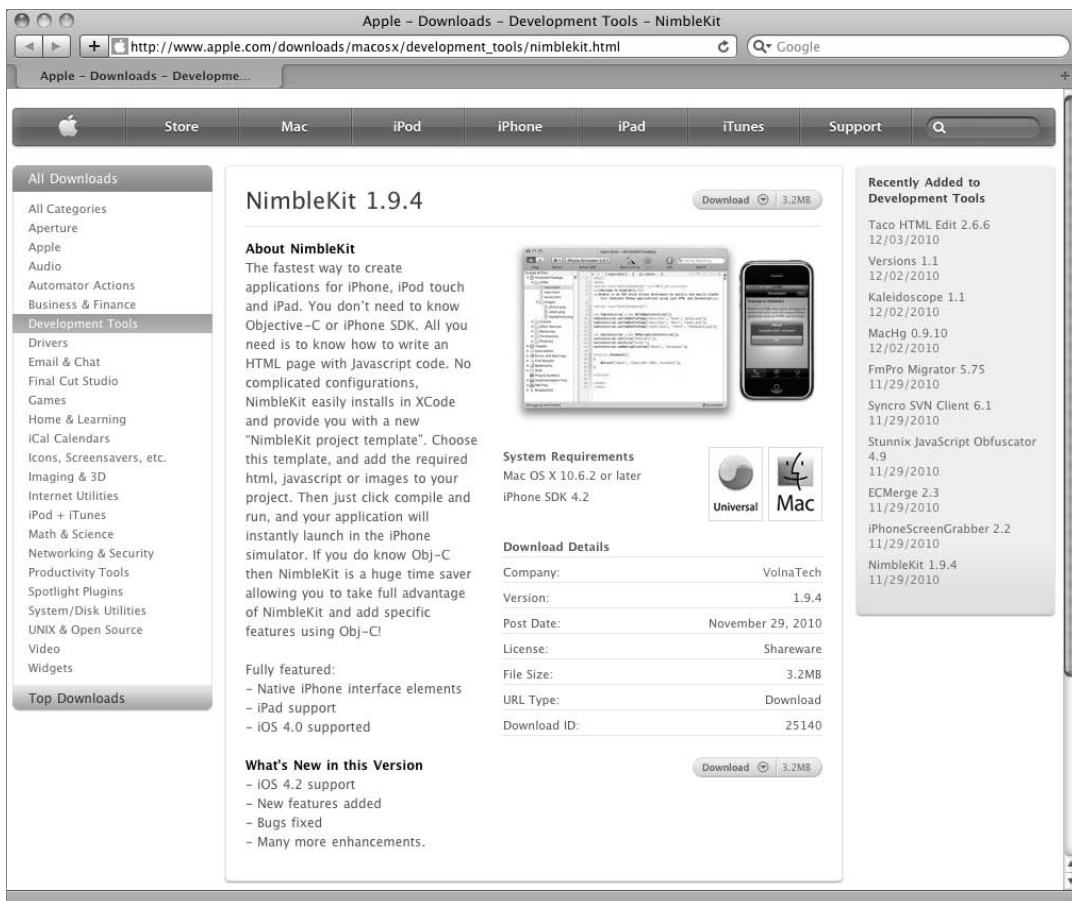
Tak, to prawda. Nie znaczy to jednak, że musisz samodzielnie tworzyć jakikolwiek kod tego rodzaju!

W gruncie rzeczy, co by było, gdyby ktoś inny napisał go już za Ciebie?

Pomyśl o platformach JavaScript umożliwiających projektowanie treści, które mają być prezentowane za pomocą przeglądarek internetowych. Dwa z moich ulubionych przykładów — jQuery i Yahoo! User Interface (YUI) — pomagają projektantom w wykorzystywaniu możliwości kodu JavaScript i uwalniają ich od konieczności wykonania większości uciążliwej pracy programistycznej (co z kolei prowadzi do pytania o to, jak dużo musi być tego kodu). Kod odpowiedzialny za niektóre bardzo ciekawe efekty został dla nas opracowany wcześniej i wbudowany w moduły, my zaś możemy po prostu zerknąć w nie i wykorzystać odpowiednio bez konieczności pisania czegokolwiek od podstaw. W gruncie rzeczy nie powinieneś również zapominać o innym istotnym aspekcie używania platform: zawarty w nich kod został bardzo solidnie sprawdzony (i w dalszym ciągu jest regularnie testowany oraz aktualizowany), nie musimy więc też debugować własnej twórczości!

Podobnie jest i tutaj. Istnieją platformy projektowe iOS, które spełniają takie samo zadanie w przypadku tworzenia aplikacji dla iPhone'a. Platforma, z której korzystałem najczęściej, nosi nazwę NimbleKit (rysunek 1.4). To wspaniałe narzędzie oferuje sporo kodu obiektowego C, który został już dla nas przez kogoś napisany i pozwala korzystać z natywnych funkcji i zachowań systemu iOS, a także został dostosowany do wywoływania w akcjach przeprowadzanych w projektach wykorzystujących kod HTML, CSS i JavaScript. Gdy się nad tym zastanowisz, zacznie Ci to bardzo przypominać używanie tych języków do instruowania przeglądarki internetowej, aby wyświetlała określoną treść i zachowywała się we wskazany sposób. Po prostu używamy tu innej platformy kodowej i projektujemy dla konkretnego systemu operacyjnego (iOS) oraz sieci dystrybucyjnej (iTunes).

O tym właśnie będą traktować kolejne rozdziały tej książki. Poświęcę je zgłębianiu istotnych cech charakterystycznych interfejsów i zachowań iOS oraz prezentacji tego, jak NimbleKit radzi sobie z pełnieniem roli mostu łączącego dobrze znane Ci języki projektowania na potrzeby WWW i nowe urządzenia (oraz ich własne, natywne języki programowania). Następnie, po przyjrzeniu się kilku przykładom projektowania aplikacji iOS opierających się na treści, zajmiemy się tym, jak zgłosić je do firmy Apple, aby zostały przez nią zatwierdzone, a także tym, jak rozpocząć ich dystrybucję lub sprzedaż w sklepie iTunes.



Rysunek 1.4. NimbleKit jest narzędziem do tworzenia aplikacji polecanym na stronie internetowej firmy Apple

Pod przywództwem Apple trwającym od roku 2001 po dzień dzisiejszy możemy obserwować, jak treści tego rodzaju wiodą prym w obecnych czasach i jak te niewielkie, proste urządzenia z intuicyjnymi i spójnymi interfejsami oraz zachowaniami dostarczają przydatnych treści — codziennych treści — użytkownikom właśnie tam, gdzie się znajdują, i wtedy, gdy są im potrzebne. Wszystko, co powinniśmy zrobić, sprowadza się do skupienia się na ludziach i ich prawdziwych potrzebach, przypomnienia sobie naszych umiejętności w zakresie projektowania na potrzeby WWW, poznania pewnej liczby nowych wskazówek i sztuczek oraz doczepienia naszego wagonu do święcącego niesamowite triumfy pociągu Apple.

Właśnie zaczynamy to robić!

Podsumowanie

Oto, czego się dowiedziałeś w tym rozdziale:

- ▶ Nie daj się zwieść nazwie „iPhone”. W rzeczywistości urządzenie to jest kieszonkowym komputerem wyposażonym w aplikację Telefon. Oznacza to, że może ono robić cudowne rzeczy, a my możemy pomóc niektóre z nich zrealizować!
- ▶ Wprowadzenie całego ekosystemu iOS oznaczało niesamowitą zmianę strategii firmy Apple, która zaczęła się koncentrować wokół stylu życia, wszechobecnej informacji i treści. To zaś otworzyło ogromne możliwości przed ludźmi zajmującymi się projektowaniem różnych cyfrowych rzeczy.
- ▶ System iOS oferuje bardzo intuicyjny, dotykowy interfejs użytkownika i w pełni integruje się z całą resztą naszego życia, nie tylko z jego częścią zawodową. Jak możemy projektować aplikacje wpisujące się w ten kontekst?
- ▶ Natywne aplikacje są programowane w obiektowym C, nie oznacza to jednak, że projektanci muszą się uczyć tego języka. Mogą oni współpracować z programistami lub wykorzystywać platformy kodu, aby osiągnąć swoje cele. W książce tej zajmiemy się tym ostatnim.

Skorowidz

A

- Active
 - Configuration, 63
 - Executable, 145
- Add
 - Certificate, 214
 - New Application, 223
 - to Project, 51, 78
- addButton, 128
- addFlexibleSpace, 129
- aktualizacja
 - status, 90
 - treści, 90
- Anderson Erin, 236
- Android, 181, 183, 186, 189, 191
- Any iOS, 61, 63
- API, 106
 - Google Maps, 106
- aplikacja
 - aktualizacja, 207
 - budowanie, 62
 - czas zatwierdzenia, 224
 - data udostępnienia, 224
 - długość nazwy, 48
 - dobrze zaprojektowana, 24
 - dodawanie, 223
 - dołączanie grafiki, 50
 - dystrybucja, 62
 - funkcjonalność, 242
 - identyfikator, 217
 - ikona, 48, 51, 228
 - informacje, 223, 224
 - kanał mediów społecznościowych, 204
 - kategoria, 227
 - komunikacja
 - marketingowa, 207
 - z użytkownikami, 209
 - konserwacja treści, 235
 - mobilna, 21
 - platformy do tworzenia, 191
 - nazwa, 46, 48, 224
 - pakiet, 47
 - planowanie, 237
 - schemat, 240
 - serwis WWW, 202
 - słowa kluczowe, 228
 - strona wizualna, 241
 - studio projektowania, 30
 - testowanie, 56, 212
 - tworzenie, 36
 - tytuł, 71
 - uniwersalna, 43, 141
 - uruchamianie, 62
 - zarządzanie, 223
 - zmiana nazwy, 47
 - zrzuty ekranów, 228
- App
 - ID, 217, 218
 - Information, 223
 - Marketing
 - and Identity Guidelines for Developers, 199
 - Artwork License Agreement, 198
 - Name, 224
- App Store, 194
 - czcionki dozwolone, 201
 - liczba aplikacji, 201
 - możliwości marketingowe, 197
 - oceny i komentarze, 236
 - opis aplikacji, 200
 - używanie terminu, 201

Appcelerator, 186
 AppControls, 174, 175
 AppDelegate.m, 62
 apple styl predefiniowany, 183, 184
 Apple
 Computer, Inc., 19
 Developer
 rejestracja jako firma, 195
 rejestracja jako klient, 196
 rejestracja jako przedsiębiorstwo, 196
 rejestracja pod nazwiskiem własnym, 194
 Developer ID, 30, 31, 33
 Development Agreement, 33
 ID, 32
 klient, 31
 światowa konferencja twórców, 33
 twórca, 31
 Worldwide Developer Relations Certification Authority, 215
 znaki towarowe, 34
 Application Loader, 229, 231
 arkusz
 ostrzeżenia, 90
 stylu, selektor, 151
 article, 152, 153
 aside, 153
 Assign a New Apple Device, 219
 audio, 126, 131
 odtwarzacz, 126
 audiocontrols, 128
 Automatic Profile Selector/iPhone
 Developer, 61
 Distribution, 63
 Availability Date, 224
 Available on the App Store, 34, 198
 zasady korzystania, 198

B

Base SDK, 61, 63
 Based On, 61, 63
 bieżąca lokalizacja użytkownika aplikacji, 110
 BlackBerry, 181
 blogger.js, 91, 94
 błąd budowania aplikacji, 64
 border-radius, 156

box-shadow, 139
 budowanie aplikacji, 62
 błąd, 64
 Build, 61, 63, 64
 and Run, 146
 Bundle ID, 224
 button, 115
 buttonPressed, 129

C

canvas, 154
 Certificate Signing Request, 213, 230
 certsSigningRequest, 214
 certyfikat
 dystrybucyjny, 213, 215, 216, 221
 nowy, 217
 pośredni WWDR, 215
 problemy, 217
 rozwojowy, 213, 216
 stary, 217
 twórcy, 59
 wniosek, 213
 class, 151
 Clean All Targets, 64
 Code Signing Identity, 61, 63
 color-stop, 172, 174
 Configuration, 63, 65, 66
 controls, 126, 134
 Copy items into destination group's folder (if needed), 51
 Create a New App ID, 217
 CSR, 213, 230
 CSS, 25
 gradient, 170
 CSS3, 155, 174, 189
 AppControls, 174
 border-radius, 97
 kod, 175
 kontrolki, 175
 narzędzia, 174
 CSS3 Please!, 177
 czas
 okalny, 69
 zatwierdzenia aplikacji, 224
 czcionka, 164, 167

D

Dashcode, 36
 data udostępnienia, 224
 Debug, 61
 debugowanie, 59
 Developer Program, roczny koszt udziału, 196
 Development, 217
 development certificate, 215
 Development Provisioning Assistant, 59, 62, 217, 218
 Device, 63
 diagram projektu, 241
 Discount for Educational Institutions, 225
 Dist., 222
 Distribution, 63, 222
 Provisioning Profile, 64
 Distribution-iPhoneos, 64
 div, 151
 długość
 geograficzna, 109
 klucza, 214
 nazw aplikacji, 48
 dmg, 36
 dodawanie
 aplikacji, 223
 plików, 51
 znacznika, 110
 dolna część strony, 152
 dołączanie dźwięku, 128
 Dostęp do pęku kluczy, 213, 216
 dotykowy interfejs, 192
 dystrybucja
 aplikacji, 62
 metoda, 222
 dystrybucyjny profil, 222
 dzielenie strony, 245
 dźwięk, 125, 128

E

edytora, widok, 45
 efekt
 gradientu, 72
 przezroczystości, 157
 ekran, 68
 margines wokół, 139
 wymiary, 68
 zrzuty, 200

elementy sterujące, 134
 wyróżnianie, 247
 emulator iOS, 183
 End User License Agreement, 228
 Enterprise Program, 196
 etykieta przycisku, 129
 EULA, 228

F

Facebook, 204
 Favorites, 73
 Featured, 73
 Financial Reports, 223
 folder
 Classes, 46
 Frameworks, 46
 HTML, 45, 46, 51
 Other Sources, 46
 Resources, 46, 51, 52
 footer, 151, 152
 fora dla projektantów aplikacji, 212
 formatowanie przycisku HTML, 114
 fotografia ponad treścią, 102
 funkcja dodawania znacznika, 110
 funkcjonalność aplikacji, 242

G

gładzik, 23
 główny katalog serwera, 45
 Google Maps, 106, 107
 formaty łączy, 111
 górna część strony, 152
 gradient, 170, 171
 arkusz stylów, 170
 CSS, 170
 efekt, 72
 obszar ograniczony, 172
 skośny, 172
 Grafic User Interface, 19
 graficzny interfejs użytkownika, 19
 grafika
 dodawanie do projektu, 78
 dołączanie do aplikacji, 50
 przycisku, 129
 rozruchowa, 52, 53, 54

grafika
 współrzędne narożnika, 100
 zmniejszenie, 97
 grouped, 81
 GUI, 19

H

Halvorson Kristina, 233
 header, 151, 152
 height, 134
 hierarchia wizualna, 243
 hiperłącza, 247
 HTML, 25, 45
 typografia, 162
 HTML5, 126, 134, 150, 155, 189
 hybrid, 109

I

Icon file, 51
 id, 151
 IDE, 30
 Identifier, 220
 identyfikator
 aplikacji, 217
 pakietu, 224
 twórcy 32
 Apple, 30
 urządzenia, 220
 ikona
 aplikacji, 48, 51
 karty, 74
 kompletny zestaw, 50
 rozmiar, 49
 wymiary, 228
 informacje
 dla projektantów aplikacji, 212
 o aplikacji, 223
 o wersji, 225
 init, 121
 insertCategoryNamed, 81
 insertRecord, 77, 81
 instalacja
 na urządzeniu, 59
 profilu, 221
 próbna, 212

Integrated Development Environment, 30
 interfejs
 dotykowy, 192
 użytkownika, 23, 67
 iOS, 18, 22, 27, 186, 189
 emulator, 180
 Dev Center, 37, 212
 Provisioning Portal, 59, 63, 212, 222
 SDK, 30
 pobieranie, 35
 wielkość pakietu, 35
 iPad, 49, 138
 ekran, 23, 120
 iPhone, 18, 19, 27, 49, 181
 ekran, 23
 iPhone 4, 43, 49
 symulacja, 58
 iPhone SDK, 30
 iPod, 19, 20, 49
 Item 0, 51
 iTunes, 18
 App Store, 18
 Connect, 64, 197, 221

J

JavaScript, 25, 189
 Jobs Steve, 33
 jqc styl predefiniowany, 183, 184
 jqTouch, 180, 182, 183, 191
 jQuery, 25, 182

K

karta
 ikona, 74
 nawigacyjna, 73
 etykiety własne, 74
 maksymalna liczba, 74
 projektu, 239
 kategorie
 aplikacji, 227
 kart nawigacyjnych, 73
 własne, 74
 Kernest, 163
 kierunek zmiany gradientu, 172
 klawiatura, 23

klient firmy Apple, 31
 kolor
 pasek stanu, 69
 pasek tytułu, 71, 108
 próbnik, 72
 komentarze, 236
 kompletny zestaw ikon, 50
 komputer Mac, 30
 konserwacja treści aplikacji, 235
 kontrolka
 natywne, 24
 wygląd półprzezroczysty, 112
 konwencje, 244
 Krug Steve, 242

L

Laboratorium Map Google, 109
 LatLng Tooltip, 109
 licencja NimbleKit, 55, 62
 lista
 definicji, 86, 87
 numerowana, 85
 uporządkowana, 85
 logotypy Apple, 34
 lokalizacja użytkownika aplikacji, 110

Ł

łącza, 247
 łączności sprawdzanie, 91, 112

M

m4v, 134
 Mac, 30
 Mac OS X Snow Leopard, 30
 Manage Your Applications, 223
 map, 109
 mapa, 106
 osadzanie, 109
 ramka wokół widoku, 120
 szpilka, 111
 widok, 121
 mapy Google, widoki własne, 107

marginies
 dodanie, 97
 przycisku, 120
 wokół
 ekranu, 139
 elementu, 97
 marketing, 240
 meta, 108
 metoda dystrybucji, 222
 Mobile Safari, 91
 mobileprovision, 221
 mobilne aplikacje, 21
 możliwości
 dotykowe, 189
 pakietu Xcode, 42
 MP3, 126
 MPEG-4, 134

N

nagłówek, 152
 narożniki
 zaokrąglenie, 116
 obrazu, 97
 narzędzia
 CSS3, 174
 programistyczne, 36
 nav, 154
 navController, 77
 nazwa
 aplikacji, 46, 47, 48
 długość, 48
 ekranowa, 48
 iTunes, 48
 profilu, 221, 222
 projektu, 43
 strony, 108
 niezależność renderowanej typografii HTML, 162
 NimbleKit, 25, 26, 45, 56, 76, 191
 instalowanie, 38
 licencja, 39, 55
 numer seryjny licencji, 62
 pobieranie, 37, 38
 wersja licencjonowana, 62
 Nitobi, 180

NKAlert, 90, 112
 NKAudioPlayer, 128, 131
 NKButton, 107, 108, 114
 NKImage, 78
 NKImageView, 98, 99, 100
 NKIsInternetAvailableViaCellularNetwork, 91
 NKIsInternetAvailableViaWifi, 91
 NKIsPageSupportsAutoOrientation, 136, 142
 NKit, 61, 63
 NKMapView, 109, 110, 114, 117, 123
 NKNavigationController, 71, 136, 142
 NKOpenURLInSafari, 91
 NKTableViewCell, 76, 87
 NKToolBar, 99, 128, 131
 NKVideo, 142
 NKVideoPlayer, 140, 142
 numer
 seryjny licencji NimbleKit, 62
 SKU, 224
 numeracja wersji, 54

O

obiektowy język C, 42
 obraz
 narożniki zaokrąglone, 97
 ponad treścią, 98
 urządzeń
 iOS, 200
 kolor tła, 200
 wbudowany, 95
 wpleciony, 95
 zastępczy, 134
 obrót (działanie), 189
 oceny, 236
 odtwarzacz
 audio, 126, 131
 wideo, 144
 odtwarzanie
 pliku
 audio, 126
 wideo, 134
 ponowne, 129
 wznowienie, 129
 onClick, 117
 openInMaps, 111

osadzanie
 mapy, 109
 pliku wideo, 146

P

pakiet
 aplikacji, 47
 identyfikator, 224
 wersja, 54
 Palm OS, 181
 panel
 grup i plików, 44
 widoku
 edytora, 44
 szczegółów, 44
 parametry projektu, 239
 pasek
 kart, 72
 nawigacyjnych, wymiary, 75
 stanu, 108, 114, 120
 iOS, 68, 69
 kolor, 69
 wymiary, 70
 tytułu, 71, 108, 114, 120
 kolor, 71, 108
 wymiary, 70
 pauza, 129
 PhoneGap, 180, 183, 186, 187, 191
 zalety, 181
 planowanie aplikacji, 237
 platformy do tworzenia aplikacji mobilnych, 191
 play, 129
 plik
 audio, 126
 celu, 60
 CSS, 45
 dmg, 36
 graficzny, 45
 dodawanie do projektu, 78
 kompresowanie, 64
 plist, 52
 projektu, 60
 wideo, 134, 146
 odtworzenie, 134
 wykonywalny
 budowanie, 64

podstawowe informacje na temat aplikacji, 224
 podsumowanie ról, 240
 podwójne puknięcie (działanie), 189
 połączenie internetowe, 90
 ponowne odtwarzanie, 129
 poster, 134
 powrót na górę strony, 69
 poziom

- ceny, 225
- przezroczystości, 112

 prekompilowany nagłówek, 47
 Price Tier, 225
 Primary Category, 228
 proces konserwacji treści, 235
 profil

- dystrybucyjny, 222
- instalowanie, 221
- nazwa, 221, 222
- testowy, 220

 Project, 78
 projekt

- diagram, 241
- dodawanie plików, 51
- karta, 239
- nazwa, 43
- podstawowe parametry, 239
- problemy, 238
- rozwój, 240
- zmiana nazwy, 47

 projektant, typy osobowości, 232
 projektowanie

- wytyczne, 231
- aplikacji, 21, 22, 30

 prosty widok tabeli, 75
 Provisioning, 222

- Portal, 62

 próbna instalacja, 212
 Próbnik kolorów, 72
 przejrzystość, 161
 przezroczystość, 157
 przycisk

- etykieta, 129
- formatowanie, 116
- grafika, 129
- HTML, 114, 115
- marginesy, 120

odtwarzania, 131
 ponownego odtwarzania, 131
 wstrzymania odtwarzania, 131

R

ramka wokół widoku mapy, 120
 Ready to Load Binary, 229
 Red-Green-Blue-Alpha, 157
 reguła @font-face, 162
 Release, 63
 Resources, 51

- folder, 52

 resume, 129
 RGBA, 157
 Rights and Pricing, 224
 rola

- podsumowanie, 240
- przydział wykonawcom, 238

 rozmiar ikony, 49
 rozruchowa grafika, 52, 53
 rozszerzenie

- framework, 46
- h, 46
- m, 46
- pch, 47
- plist, 47, 51, 54
- xcodeproj, 47
- xib, 46

 rozwój projektu, 240
 RSA, 214

S

Sales and Trends, 223
 satellite, 109
 Scalable Vector Graphics, 163
 schemat aplikacji, 240
 SDK, 63

- bieżące wydanie, 36
- wersja beta, 36

 Secondary Category, 228
 section, 150, 153
 selektor arkusza stylu, 151
 Sencha Touch, 189, 191
 setDisplayRegion, 110

Simulator, 36, 56, 64, 137
 uruchamianie, 56
 skompresowanie pliku, 64
 SKU Number, 224
 słowa kluczowe aplikacji, 228
 Spotlight, 49
 sprawdzanie połączenia internetowego, 91
 src, 134
 standard numeracji wersji, 54
 sterujące elementy, 134
 stopka, 152
 strategia tworzenia treści, 233, 237
 strona
 część
 dolna, 152
 górną, 152
 dzielenie, 245
 nazwa, 108
 studio projektowania aplikacji, 30
 SVG, 163
 Symbian, 181
 symulator
 iPada, 58
 iPhone'a, 57, 58
 szerokość
 widoku tabeli, 82
 geograficzna, 109
 szum wizualny, 248

Ś

środek widoku mapy, 109

T

tableView, 76
 tekst
 cień, 116
 uwypuklenie, 116
 testowanie
 aplikacji, 36, 56, 212
 na urządzeniu, 59
 testowy profil, 220
 Titanium
 Developer, 187
 Mobile, 186, 191

Top Rated, 73
 treść
 aktualizowanie, 90
 konserwacja, 235
 koszty, 234
 strategia tworzenia, 233, 237
 wielkość porcji informacji, 234
 zarządzanie, 94, 236
 Twitter, 90, 95, 236
 tworzenie
 aplikacji, 36
 treści, 233
 twórca, 31
 korzyści, 33
 ograniczenia i poufność, 33
 umowa, 33
 typografia HTML, 162
 tytuł aplikacji, 71

U

UDID, 220
 UI, 23
 elementy własne, 24
 układ ekranu, 107, 114, 120
 umowa
 licencyjna
 Apple, 198
 użytkownika końcowego, 228
 twórcy Apple, 33
 uniwersalna aplikacja, 141
 uruchamianie aplikacji, 62
 User Interface, 23
 ustawienia, 60, 61, 62
 budowania, 60
 plik
 celu, 64
 projektu, 64

V

Version Information, 225
 video, 134, 135
 VolnaTech, 191

W

wbudowane kategorie kart nawigacyjnych, 73
 -webkit-, 139
 wersja
 numeracja, 54
 pakietu, 54
 wiadomości, źródło, 95
 wideo, 133, 146
 odtworacz, 144
 odtworanie, 134
 widok
 edytora, 45
 Google Maps, 107
 mapy, 121
 środek, 109
 tabeli, 75, 87, 88, 89
 liczba wierszy wyświetlanych, 77
 prosty, 75
 szerokość, 82
 wysokość wiersza, 78
 zgrupowany, 80
 width, 134
 Windows, 30
 Windows Mobile, 181
 wizualna strona aplikacji, 241
 wizualny szum, 248
 wniosek o certyfikat, 213
 Worldwide Developer Conference, 33
 wskaźnik
 mocy sygnału telefonii komórkowej, 68
 poziomu naładowania akumulatora, 69
 współrzędne
 geograficzne miejsca, 110
 narożnik grafiki, 100
 WWDC, 33
 WWDR, 215
 wykonawcy
 przydział ról, 238
 zakres odpowiedzialności, 240
 wymiary
 ekranu, 68
 paska tytułu, 72
 widoku, 109
 wyróżnianie elementów sterujących, 247
 wznowienie odtwarzania, 129

X

Xcode, 30, 36, 42, 51
 Active
 Architecture, 57
 Configuration, 56, 60
 Executable, 57, 60
 Build and Run, 57
 Build, 61
 Choose, 43
 Comments, 61
 Configurations, 61
 Debug, 56
 Detail View, 44
 Device, 58, 60
 Editor View, 44
 File, 42
 General, 61
 Get Info, 60
 Groups & Files, 44
 Hardware, 58
 Info, 60
 iPad Simulator 4.3, 58
 iPad, 43
 iPhone (Retina), 58
 iPhone Simulator 4.3., 58
 iPhone, 43
 iPhone/iPad, 43
 możliwości pakietu, 42
 New Project, 42
 okno główne, 44
 Overview, 56
 Product, 43
 Project Info, 61
 Project, 47
 Rename, 47
 Scale, 58
 typ aplikacji, 43
 Window, 58
 xcodeproj, 44

Y

Yahoo! User Interface, 25
 YUI, 25

Z

zaokrąglenia narożników
 elementu, 157
 obrazu, 97

zarządzanie
 aplikacjami, 223
 treścią, 94, 236

zatwierdzenie aplikacji, 224

zdjęcia
 ekranów, 200
 urządzeń udostępniane przez Apple, 34

zestaw ikon, 50

zgrupowany widok tabeli, 80

zintegrowane środowisko programistyczne, 30

zmiana nazwy
 aplikacji, 47
 projektu, 47

zmniejszenie grafiki, 97

znacznik
 dodawanie, 110
 położenia geograficznego, 109

znaki towarowe Apple, 34

zrzut ekranu, 200
 wymiary, 228

Ż

źródło wiadomości, 95

◀ **Są z nami praktycznie wszędzie.** Podczas zakupów, jazdy samochodem, w trakcie joggingu... a nawet na basenie. Zmieniły nasz sposób pozyskiwania informacji, poszukiwania rozrywki i komunikacji ze światem. Dzięki urządzeniom takim jak iPhone czy iPad możesz w jednej chwili sfotografować niezwykłą scenę i rozesłać zdjęcie wszystkim znajomym, by potem wspólnie wymieniać na jego temat opinie, jednocześnie sprawdzając prognozę pogody i to, jak daleko jesteś od celu, do którego zmierzasz... Możliwości mobilnych aplikacji pracujących pod kontrolą iOS ogranicza dziś jedynie ludzka wyobraźnia! Jednak nic nie przemawia do niej tak, jak twarde dane: tylko do połowy 2010 roku sprzedanych zostało ponad 85 milionów urządzeń pracujących w tym systemie, a w iTunes App Store dostępnych było ponad 250 tysięcy aplikacji... pobranych ponad 15 miliardów razy!

◀ Oto książka poświęcona projektowaniu atrakcyjnych, zgodnych ze standardami sieciowymi aplikacji działających pod kontrolą systemu iOS – aplikacji, które można udostępniać lub sprzedawać za pośrednictwem iTunes App Store. Podręcznik stanowi doskonałe wprowadzenie w świat kodu HTML, stylów CSS i języka Java Script, z którymi zaprojektujesz niezawodne natywne aplikacje, działające na iPhone'ach, iPodach touch oraz iPadach. Dzięki niemu nauczysz się korzystać z platformy obiektowego C (o nazwie NimbleKit). Platforma ta jest kolekcją bibliotek elementów umożliwiających tworzenie programów za pomocą obiektowego języka C bez konieczności samodzielnego pisania jakiegokolwiek kodu. Książka ta jest ponadto wszechstronnym przewodnikiem po kwestiach wizualizowania, planowania, tworzenia i dystrybucji aplikacji iOS oraz szczegółową instrukcją projektowania rozmaitych aplikacji.

◀ **Z tej niezwyklej książki dowiesz się m.in., jak samodzielnie:**

- pobrać i zainstalować bezpłatne oprogramowanie do tworzenia aplikacji iOS
- używać środowiska Xcode do zarządzania zasobami projektu, testowania aplikacji i pakowania plików binarnych
- implementować standardowe elementy i działania iOS w celu zapewnienia użytkownikom urządzeń firmy Apple interfejsu zgodnego z ich oczekiwaniami oraz przyzwyczajeniami
- ulepszać projekty swoich aplikacji dzięki możliwościom oferowanym przez języki HTML5 i CSS3
- budować solidne podstawy do tworzenia aplikacji WWW, które mogą być używane na innych smartfonach i urządzeniach mobilnych
- projektować treści tekstowe, graficzne, audio i wideo, przeznaczone do prezentacji na sprzęcie mobilnym

Kristofer Layon to projektant serwisów WWW i aplikacji oraz wykładowca. Jego firma, Aesthete Software, projektuje aplikacje mobilne dla klientów działających w różnych branżach, w tym medycznej, fotograficznej i edukacyjnej. Kristofer jest autorem znanego bloga na temat projektowania. Występuje także na licznych konferencjach poświęconych projektowaniu aplikacji i serwisów WWW, a ponadto uczy programistów tworzenia aplikacji iOS podczas autorskich warsztatów Two Apps Per Day.

helion.pl
księgarnia
internetowa

Nr katalogowy: 6847



Księgarnia internetowa:
<http://helion.pl>



Zamówienia telefoniczne:
0 801 339900



0 601 339900



Helion

Sprawdź najnowsze promocje:
• <http://helion.pl/promocje>
Książki najchętniej czytane:
• <http://helion.pl/bestsellery>
Zamów informacje o nowościach:
• <http://helion.pl/nowości>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>



ISBN: 978-83-246-3331-9



9 788324 633319

Cena: 49,00 zł

Informatyka w najlepszym wydaniu